CMPE 124 Lab 5: A 4-bit Unsigned Comparator

Leonardo Blas

***Abstract*—The purpose of this lab project is to implement a 4-bit unsigned comparator with the help of our previously implemented 74LS163 counter.**

I. Introduction

This lab project is based on the implementation of a 74LS163 counter, a logical circuit that tracks time by using 8 bits.
We use this clocks' input to count from 0 to 15 and compare these values with a constant 4-bit number, generated with 4 binary switches.

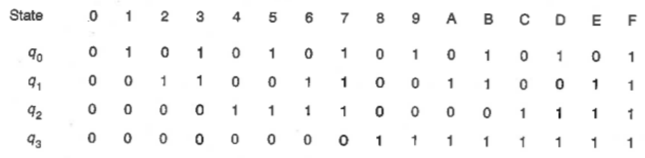
II. Design Methodology

For this lab project, we use the 74LS163 4-bit counter. This 4-bits counter requires not only a voltage source—to function—and a clock—to provide it with time input—but also a binary switch, to reset it and ensure the wave functions it outputs are correct. Additionally, although we use a resistor, its value can be changed or ignored; our 1kΩ resistor is part of the circuit to demonstrate that we are not focusing on the input voltage, but on the H and L states instead.
Furthermore, we create a constant 4-bit number by implementing 4 binary switches, and we then implement a circuit to compare this number with the counter's variable values. Finally, to verify our circuit, we check its functionality with a hex display connected to our counter and LED lights connected to the circuit's output (that turn on when the counter's output is less, greater, or equal to the switches' value).

A. Parts List

- Clock
- 5V Source
- A 74LS163 counters
- Binary switches
- Resistor
- AND gates
- OR gates
- NOT gates
- LED lights
- Hex display

B. Truth Tables



State	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
q_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
q_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
q_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
q_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Figure 1. Truth table for a 74LS163 4-bit counter.

C. Schematics

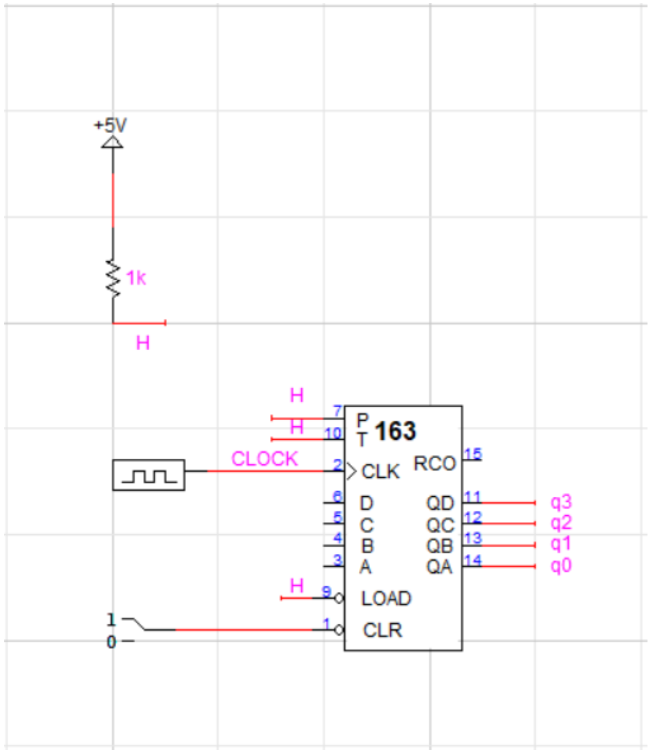


Figure 2. 74LS163 clock-counter circuit scheme.

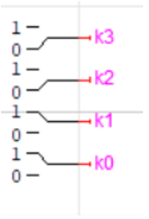


Figure 3. Setting the switches to encode for the decimal number 3.

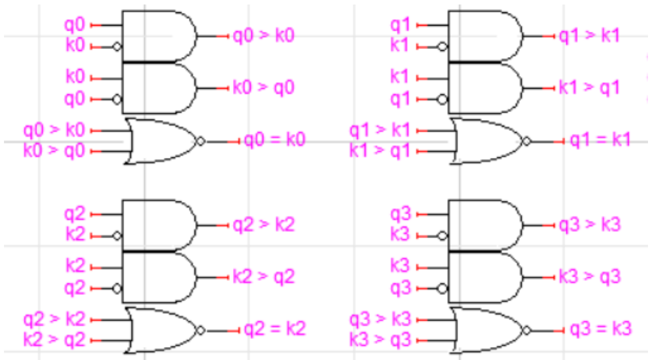


Figure 4. Comparing individual switches with their clock bits counterparts.

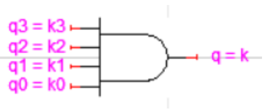


Figure 5. Checking if the counter's value is equal to the switches' value.

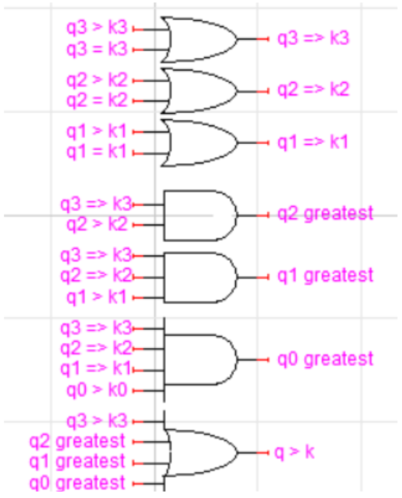


Figure 6. Checking if the counter's value is greater than the switches' value.

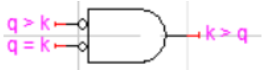


Figure 7. Checking if the counter's value is less than the switches' value.

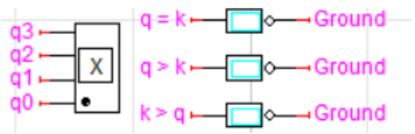


Figure 8. A hex display and three LEDs, to test our circuit's results.

III. Testing Procedures

1. Create a counter as pictured in Figure 2.
2. Set up the binary switches as pictured in Figure 3.
3. Set up the logic gates and their input as pictured in Figures 4, 5, 6, and 7.
4. Set up the testing circuit, as pictured in Figure 8.
5. Reset the clock with a switch and run the simulation.

IV. Testing Results

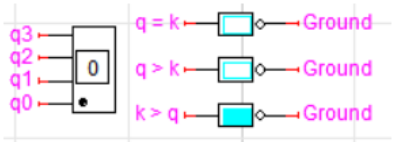


Figure 9. Testing circuit when the counter's value 0.

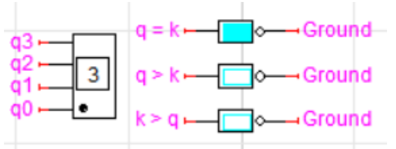


Figure 10. Testing circuit when the counter's value is the decimal value 3.

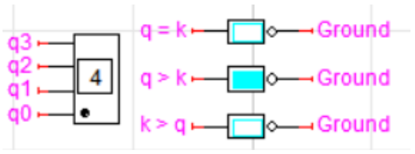


Figure 11. Testing circuit when the counter's value is the decimal value 4.

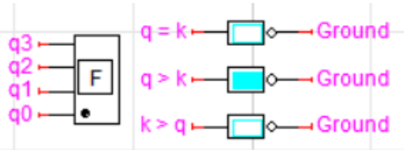


Figure 12. Testing circuit when the counter's value is the decimal value 15.

V. Conclusion

In conclusion, a 4-bit binary number can be represented by a 74LS163 clock-counter circuit, as well as four binary switches. We can process these binary values via logic gates—just like we did in our previous labs—and create a circuit that compares the counter's output with the switches' constant value. By implementing this comparator circuit, we've come closer to learning about some of the most fundamental parts of computers, like arithmetic logic units in this case.